Karan Yuvraj Singh 19BLC1097

```
library(e1071) library(caret)

data=read.csv("C:/Users/91828/Downloads/Cleaned-Data.csv",

header=TRUE)

# taking first 1000 rows
data=data[0:1000,]

# remove Na values
data=na.omit(data)
str(data)

## 'data.frame':     1000 obs. of 27 variables:
## $ Fever   : int 1 1 1 1 1 1 1 1 1 1 ... ## $ Tiredness    :
int 1 1 1 1 1 1 1 1 1 1 ...
## $ Dry.Cough    : int 1 1 1 1 1 1 1 1 1 1 ... ## $
Difficulty.in.Breathing: int 1 1 1 1 1 1 1 1 1 1 ... ## $
Sore.Throat : int 1 1 1 1 1 1 1 1 1 1 ... ## $ None_Sympton :
int 0 0 0 0 0 0 0 0 0 0 ... ## $ Pains    : int 1 1 1 1 1 1 1
1 1 1 ... ## $ Nasal.Congestion     : int 1 1 1 1 1 1 1 1 1 1
... ## $ Runny.Nose     : int 1 1 1 1 1 1 1 1 1 1 ... ## $
Diarrhea    : int 1 1 1 1 1 1 1 1 1 1 ... ## $
None_Experiencing : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Age_0.9
      : int 1 1 1 1 1 1 1 1 1 1 ... ## $ Age_10.19    : int 0
0 0 0 0 0 0 0 0 0 ...
## $ Age_20.24                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Age_25.59    : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Age_60.
      : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Gender_Female      :
int 0 0 0 0 0 0 0 0 0 0 ... ## $ Gender_Male    : int 1 1 1 1
1 1 1 1 1 1 ... ## $ Gender_Transgender   : int 0 0 0 0 0 0 0
0 0 0 ... ## $ Severity_Mild  : int 1 1 1 0 0 0 0 0 0 0 ...
## $ Severity_Moderate  : int 0 0 0 1 1 1 0 0 0 0 ... ## $
Severity_None     : int 0 0 0 0 0 0 0 0 0 1 ... ## $
Severity_Severe   : int 0 0 0 0 0 0 1 1 1 0 ... ## $
Contact_Dont.Know : int 0 0 1 0 0 1 0 0 1 0 ... ## $
Contact_No  : int 0 1 0 0 1 0 0 1 0 0 ... ## $ Contact_Yes  :
int 1 0 0 1 0 0 1 0 0 1 ...
## $ Country                  : chr "China" "China" "China" "China" ...

attach(data)
set.seed(101)

# Logistic Regression
```

```
## For Mild Severity
fit=glm(Severity_Mild~Dry.Cough+Difficulty.in.Breathing,data=data,famil
y=binomial) summary(fit)

##
## Call:
## glm(formula = Severity_Mild ~ Dry.Cough + Difficulty.in.Breathing,
##     family = binomial, data = data)
##
## Deviance Residuals:
##     Min        1Q Median        3Q      Max
## -0.7753 -0.7585 -0.7585 1.6423 1.6651
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.09861     0.14213 -7.729 1.08e-14 ***
## Dry.Cough                  0.05064     0.21264 0.238     0.812
## Difficulty.in.Breathing -0.05064 0.18738 -0.270 0.787 ## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1129 on 999 degrees of freedom
## Residual deviance: 1129 on 997 degrees of freedom
## AIC: 1135
##
## Number of Fisher Scoring iterations: 4

# making predictions using logistic regression model
probab=predict(fit,type="response")

pred=ifelse(probab>0.5,1,0)

table(pred,Severity_Mild)

##     Severity_Mild
## pred 0 1 ##
      0 748 252

mean(pred==Severity_Mild)

## [1] 0.748

## training subset
train=(Difficulty.in.Breathing==0)
trainingset=data[!train,]
trainingset_severityMmild=Severity_Mild[train]

## fit 2 using training set
fit2=glm(Severity_Mild~Dry.Cough+Difficulty.in.Breathing,data=data,fami
ly=binomial,subset=train)
probab2=predict(fit2,type="response")
pred2=ifelse(probab2>0.5,1,0)
table(pred2,trainingset_severityMmild)

##      trainingset_severityMmild
## pred2 0 1 ##
      0 352 120

mean(pred2==trainingset_severityMmild)

## [1] 0.7457627

######################## SVM #######################

#loading data again
data2=read.csv("C:/Users/ishiv/Downloads/Cleaned-Data.csv", header=TRUE

# taking first 1000 rows
data2=data2[0:1000,]

# remove Na values
data2=na.omit(data2)
str(data2)

## 'data.frame':    1000 obs. of 27 variables:
## $ Fever   : int 1 1 1 1 1 1 1 1 1 1 ... ## $ Tiredness    :
int 1 1 1 1 1 1 1 1 1 1 ...
## $ Dry.Cough    : int 1 1 1 1 1 1 1 1 1 1 ... ## $
Difficulty.in.Breathing: int 1 1 1 1 1 1 1 1 1 1 ... ## $
Sore.Throat : int 1 1 1 1 1 1 1 1 1 1 ... ## $ None_Sympton :
int 0 0 0 0 0 0 0 0 0 0 ... ## $ Pains    : int 1 1 1 1 1 1 1
1 1 1 ... ## $ Nasal.Congestion     : int 1 1 1 1 1 1 1 1 1 1
... ## $ Runny.Nose     : int 1 1 1 1 1 1 1 1 1 1 ... ## $
Diarrhea    : int 1 1 1 1 1 1 1 1 1 1 ... ## $
None_Experiencing : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Age_0.9
      : int 1 1 1 1 1 1 1 1 1 1 ... ## $ Age_10.19    : int 0
0 0 0 0 0 0 0 0 0 ...
## $ Age_20.24                : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Age_25.59    : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Age_60.
      : int 0 0 0 0 0 0 0 0 0 0 ... ## $ Gender_Female      :
int 0 0 0 0 0 0 0 0 0 0 ... ## $ Gender_Male    : int 1 1 1 1
1 1 1 1 1 1 ... ## $ Gender_Transgender   : int 0 0 0 0 0 0 0
0 0 0 ... ## $ Severity_Mild  : int 1 1 1 0 0 0 0 0 0 0 ...
## $ Severity_Moderate  : int 0 0 0 1 1 1 0 0 0 0 ... ## $
Severity_None     : int 0 0 0 0 0 0 0 0 0 1 ... ## $
Severity_Severe   : int 0 0 0 0 0 0 1 1 1 0 ... ## $
Contact_Dont.Know : int 0 0 1 0 0 1 0 0 1 0 ... ## $
Contact_No  : int 0 1 0 0 1 0 0 1 0 0 ...
```

)

```
## $ Contact_Yes             : int 1 0 0 1 0 0 1 0 0 1 ...
## $ Country                 : chr "China" "China" "China" "China" ...
```

```
## SVM
svm_train <- sample(c(TRUE, FALSE), nrow(data2), replace = T, prob = c
(0.7,0.3))

# splitting in test and train set
train2 <- data2[svm_train, ]
testing <- data2[!svm_train, ]

x=train2[c('Dry.Cough','Difficulty.in.Breathing')]
y=train2$Severity_Mild
data2=data.frame(x=x,y=as.factor(y))

# using linear kernel
tune.out=tune(svm, y ~., data=data2 ,kernel='linear', ranges = list(cos
t=c(0.001,0.1,1,10,100))) summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
## cost
## 0.001
##
## - best performance: 0.2383216
##
## - Detailed performance results:
##    cost     error dispersion
## 1 1e-03 0.2383216 0.05323898
## 2 1e-01 0.2383216 0.05323898
## 3 1e+00 0.2383216 0.05323898
## 4 1e+01 0.2383216 0.05323898
## 5 1e+02 0.2383216 0.05323898
```

```
# getting the best model
bestmodel=tune.out$best.model
summary(bestmodel)
```

```
##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = data2, ranges = list
(cost = c(0.001,
##     0.1, 1, 10, 100)), kernel = "linear")
##
```

##

```
## Parameters:
##     SVM-Type: C-classification
## SVM-Kernel: linear
##         cost: 0.001
##
## Number of Support Vectors: 340
##
## ( 170 170 )
##
##
## Number of Classes: 2
##
## Levels:
## 0 1
```

```
# plot
plot(bestmodel, data2)
xt=testing[c('Dry.Cough','Difficulty.in.Breathing')]
yt=testing$Severity_Mild
tdata=data.frame(x=xt,y=as.factor(yt))
ypred=predict(bestmodel,tdata)
confusionMatrix(ypred,tdata$y)
```

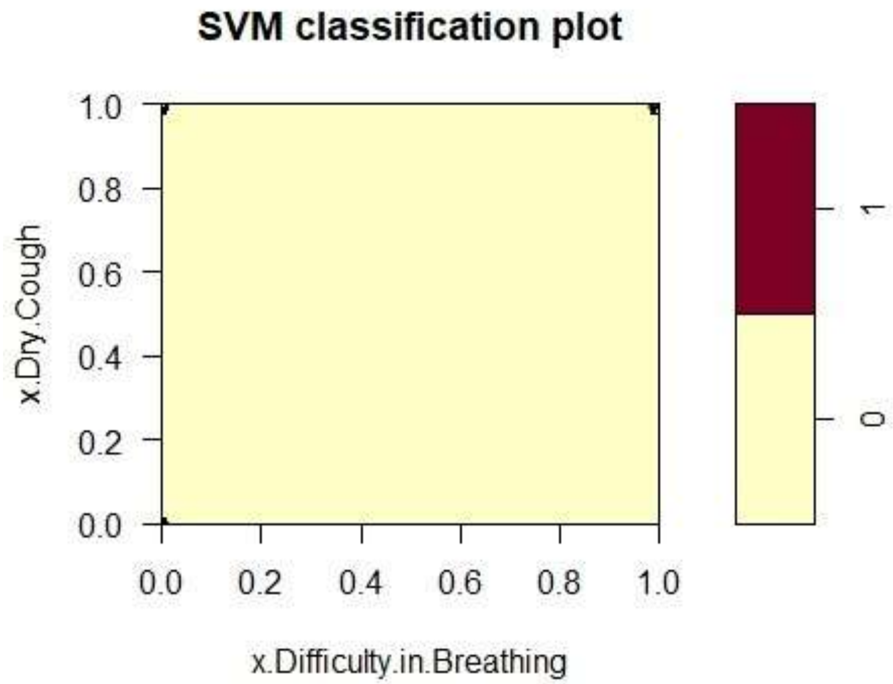
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 205 82
##           1 0 0
##
##                Accuracy : 0.7143
##                  95% CI : (0.6582, 0.7658)
##     No Information Rate : 0.7143
##     P-Value [Acc > NIR] : 0.5297
##
##                   Kappa : 0
##
## Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##    Pos Pred Value : 0.7143 ##
      Neg Pred Value :  NaN
##              Prevalence : 0.7143
##          Detection Rate : 0.7143
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
```

```
##
##        'Positive' Class : 0
##
```

```
#radial kernel
tune.out.r1=tune(svm, y ~., data=data2 ,kernel='radial', ranges = list
(cost=c(0.001,0.1,1,10,100)),gamma=1)
summary(tune.out.r1)

##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
## cost
## 0.001
##
## - best performance: 0.2383803
##
## - Detailed performance results:
##    cost  error dispersion ## 1
1e-03 0.2383803 0.0497398
## 2 1e-01 0.2383803 0.0497398
## 3 1e+00 0.2383803 0.0497398
## 4 1e+01 0.2383803 0.0497398 ##
5 1e+02 0.2383803 0.0497398

bestmodel.r1=tune.out.r1$best.model
summary(bestmodel.r1)

##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = data2, ranges = list
(cost = c(0.001,
##     0.1, 1, 10, 100)), kernel = "radial", gamma = 1)
##
##
## Parameters:
##    SVM-Type: C-classification
## SVM-Kernel: radial
##        cost: 0.001
##
## Number of Support Vectors: 340
##
## ( 170 170 )
##
##
## Number of Classes: 2
##
## Levels:
```

```
## 0 1
```

```
plot(bestmodel.r1,data2)
```



```
xtb=testing[c('Dry.Cough','Difficulty.in.Breathing')]
ytb=testing$Severity_Mild
tdata=data.frame(x=xtb,y=as.factor(ytb))
ypred.r=predict(bestmodel.r1,tdata)
confusionMatrix(ypred.r,tdata$y)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 205 82
##            1 0 0
##
##                 Accuracy : 0.7143
##                   95% CI : (0.6582, 0.7658)
##      No Information Rate : 0.7143
##      P-Value [Acc > NIR] : 0.5297
##
##                    Kappa : 0
##
## Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 1.0000
##              Specificity : 0.0000
##    Pos Pred Value : 0.7143 ##
      Neg Pred Value :  NaN
##              Prevalence : 0.7143
```

```
##          Detection Rate : 0.7143
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##        'Positive' Class : 0
##
##
```